Towards Decentralized Waypoint Negotiation

Shawn Adams and **Matthew J. Rutherford**
Department of Computer Science
University of Denver, Denver, CO
firstname.lastname@du.edu

Abstract

Cooperative multi-agent path planning around a common location has many applications, and has received significant attention from the research community. Our research is motivated by the need for groups of autonomous vehicles or mobile robots to collaboratively plan efficient paths around shared navigational coordinates (waypoints) in a distributed and decentralized manner. Our ongoing research is focused on creating a distributed solution to Dresner and Stone's Autonomous Intersection Management problem. In the future we plan to relax the constraints of this problem, and allow more flexibility in the angles of approach and departure from a single waypoint, and also plan to consider efficient group plans for multi-waypoint routes. In this paper we briefly introduce intersection management, present preliminary results for an unstructured peer-to-peer approach to the problem, and discuss future research directions.

Introduction

Today, navigation for mobile robots and autonomous vehicles operating outdoors in an unstructured environment is most often accomplished by supplying the robot with a desired route represented by a series of GPS coordinates (waypoints) that must be followed. Robots will attempt to travel in a straight line between waypoints, but typically have the ability to deviate from this to avoid obstacles detected through the use of board sensors (e.g., camera, laser rangefinders, etc).

Given topological constraints in many operating environments of interest, it is likely that robots with different routes will share waypoints, but will be approaching and departing from them with different headings. For example, in an urban environment, the organization of streets and the baseline traffic patterns will result in certain intersections being present on many different routes. Similarly, in a non-urban environment, the land features, communication networks, and stationary obstacles will result in a similar overlap of routes.

As members of the University of Denver's Unmanned Systems Lab (DU^2SL), we are interested in developing technologies that can help teams and small groups of autonomous unmanned vehicles achieve missions in various civilian applications such as search and rescue, hazardous site investigation, and the establishment of ad hoc communication networks. In order to make solutions to these problems robust, we seek decentralized, distributed solutions as opposed to approaches that rely on centralized control. In the context of intersection management, this means that the vehicles approaching a shared waypoint must communicate and collaboratively decide on a way of deconflicting the use of the area surrounding the waypoint.

Our long-term goal is to provide a distributed algorithm and associated protocol that vehicles can use to negotiate this waypoint navigation. We believe the approach should work with vehicles with different physical capabilities (e.g., turning radius and travel velocities), and with routes that result in vehicles approaching a shared waypoint from different headings, and leaving the shared waypoint at different headings. The solution arrived at by the group should be efficient for the group, and should also allow some vehicles to have higher priority (i.e., less deviation from their desired path) when necessary.

We are in the early stages of this work, and before trying to tackle the large, unconstrained problem, we are working to develop a solution to Dresner and Stone's autonomous intersection management problem without a centralized reservation manager. In this problem, vehicles are approaching an intersection in the lanes, and have 3 options for proceeding through the intersection: turn right, turn left, or go straight. The intersection is represented as a matrix of tiles, with the constraint that each tile can only be occupied by one agent at any given time.

In this paper, we describe our initial unstructured decentralized approach whereby vehicles communicate in a peer-to-peer fashion with the other vehicles near the intersection, distribute information about their desired reservation and the reservations of other vehicles they know about, and locally decide which reservations will be honored, in which should be preempted by others based on the rules of the algorithm.

In this paper we also outline our evaluation approach which combines simulation and also evaluation of the protocols on real, physical robots operating autonomously. We perform simulations at 2 different scales: at the large-scale we use a simple discrete event simulator to evaluate the protocol with a large number of robots and a large number of different conditions; at the small-scale, we use the Webots robot simulator to achieve better physical accuracy with the motion of the robots in the scenario. With the physical vehicles, we plan to evaluate the ability of a very simple autonomous device to perform the protocol in real-time while also following the desired route.

The remainder of this document is organized as follows: in the next section we briefly highlight related work in this area. In the sections following we describe we describe our current ongoing work, more details of our evaluation environment, and some of our likely future directions.

Background

Intersection management has been a topic of study by previous researchers and has various applications. Most of us are familiar with intersections in our roadways. Most of us are also familiar with the delays and corresponding frustration commonly experienced on our roadways. The motivations for research in this area are numerous and extend beyond the typical intersection with which we are familiar. Other areas where intersection management is needed include a factory floor, a battlefield, and when monitoring an enclosed area like a border. Common intersection management techniques such as traffic-lights and stop signs are less efficient than other, more recent management techniques and are not feasible in some situations.

A principle goal of research in intersection management has been to improve efficiency of and reduce delays associated with intersections. Some research has focused on improving the operation of the traditional traffic light approach (Vogel, Goerick, and von Seelen 2000). Others have focused their research on new management techniques such as (Dresner and Stone 2004) and (Dresner and Stone 2005). The Dresner research provides an example of a reservation based intersection management policy. This research involves a centralized intersection manager and intelligent agents that communicate with the manager to reserve time in the intersection. To make a reservation, agents call ahead to the intersection manager and pass data such as their current velocity and path through the intersection. The intersection manager then grants or denies the reservation based off of its knowledge of other agents' use of the intersection at the same time.

Work in Progress

The focus of our current work is on a fully decentralized solution to the basic autonomous intersection management problem. We describe our algorithm for achieving this, and present initial results achieved through simulation.

Distributed Algorithm Description

Our distributed reservation algorithm is based on the Dresner work (Dresner and Stone 2004) cited previously. In the Dresner system, the intersection manager possesses knowledge of when cars will arrive at the intersection and how long each car needs to traverse the intersection. It uses this knowledge to grant or deny reservation requests. Our distributed algorithm removes the centralized intersection manager but keeps the concepts of an intersection and a reservation from the Dresner work.

An intersection is represented as a set of uniformly-sized tiles. Each agent constructs a local copy of the intersection using the Cartesian coordinates of the lower-left corner, the number of tiles, and the tile size. In addition to the intersection information, each agent is provided with a set of waypoints that it must traverse. Each agent uses its current velocity and waypoints to calculate when it will arrive at the intersection, what tiles it needs to traverse its waypoints, and how long it will need to travel through the intersection.

An agent cannot enter the intersection without first obtaining a reservation. A *reservation* is a guarantee that it is safe to enter the intersection and is represented as a list of tiles, an entry and exit time for each tile, an agent ID, and a sequence number. The sequence number is a monotonically increasing counter used to determine if a reservation has been updated. Each time a reservation is modified the sequence number is incremented. A reservation is modified when it is created or cancelled. Each agent is responsible for managing its own reservation.

As agents enter the simulation, they are introduced to at least one peer agent chosen randomly from the list of peers previously released into the simulation. Each agent maintains a list of peers as well as a list of reservations that it knows about. During the simulation, agents exchange messages containing peer and reservation lists. Upon receipt of a peer message, each agent merges the peer and reservation lists with its own lists thereby increasing its knowledge of other peers in the system and of other peers use of the intersection. It also replies to the peer with its own reservation and peer lists.

Information merging is a key function of the distributed reservation system. As mentioned previously, there are two types of data to merge, peer identifiers and reservations. Each agent keeps a finite list of peer identifiers that it uses to send information to. Every time an agent receives a peers peer list, it checks to see if there is room on its own peer list and adds previously unknown peers to it if there is. Peers are aged off of the list so that an agent can learn new information from new peers. It is also possible that a peer has left the intersection and consequently is no longer relevant to agents still in or approaching the intersection.

Agents also must merge reservations received from peers with their own reservations list. There are two steps to performing this merge. First, an agent looks for reservations contained in the peer list that are not contained in its own list. These reservations must be adjudicated with an agents own list to determine whether there are any conflicts. Two reservations conflict when they are for the same intersection tile and overlap in time. In cases where a conflict is found, the reservation with the smaller agent ID wins the conflict and keeps the reservation. The second step is to look for reservations that have been updated. An update occurs when an agent already has a reservation for a peer and receives a reservation for the same peer but the reservation contains a larger sequence number. In this case the agent replaces the existing reservation with the new reservation and handles any conflicts created by the change.

Preliminary Results

In order to test our distributed algorithm we developed a simulator capable of running various intersection management algorithms, including the Dresner algorithm, a traffic-light algorithm, an overpass algorithm, and our distributed reservation algorithm. The primary performance measure we used was average completion time. This is a measure of how long it takes agents to complete their traversal of the intersection. A poisson distribution is used to introduce agents into the simulation at varying probabilities. Higher probabilities produce simulations with higher density traffic. Figure 1 shows average completion time for 100 runs of an 80 robot simulation in all four algorithms. The overpass results represent the ideal performance. The traffic-light results provide something to compare the reservation based algorithms to. The centralized algorithm results are a replication of the Dresner algorithm. Our initial decentralized reservation algorithm performs approximately 35 to 45 percent better than the traffic-light performance and approximately 0 to 25 percent less efficiently than the centralized reservation algorithm.

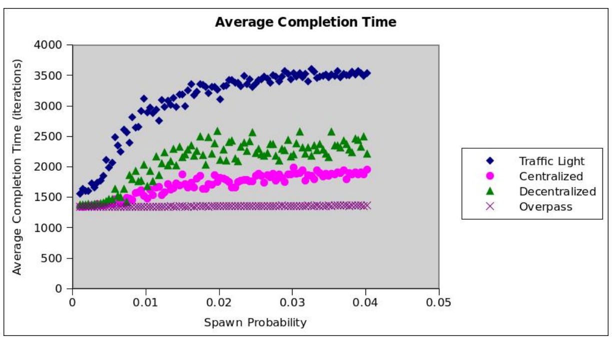

Figure 1: Total trip times for four intersection management algorithms

At lower spawn probabilities, the decentralized and centralized reservation systems perform nearly as well as the overpass system. The decentralized and centralized performance start to diverge around a spawn probability of 0.009 and the decentralized results show greater dispersion. The decentralized performance depends on how well connected the agents in the simulation become. One possible source of poor performance in the decentralized algorithm is access, or lack of, to peers' reservation information. It is possible that as peers are aged off a robot's peer list the robot stops receiving timely updates to peers' reservations which in turn causes unnecessary contention over the intersection resources and slows everyone down. We are actively investigating the nature of this performance difference; our intuition is that the two solutions should be very close as each of the robots is essentially performing the same decision making process as the centralized manager does.

Evaluation Approach

We are planning on using 3 different methodologies for evaluating our algorithm. As the initial focus of our work is

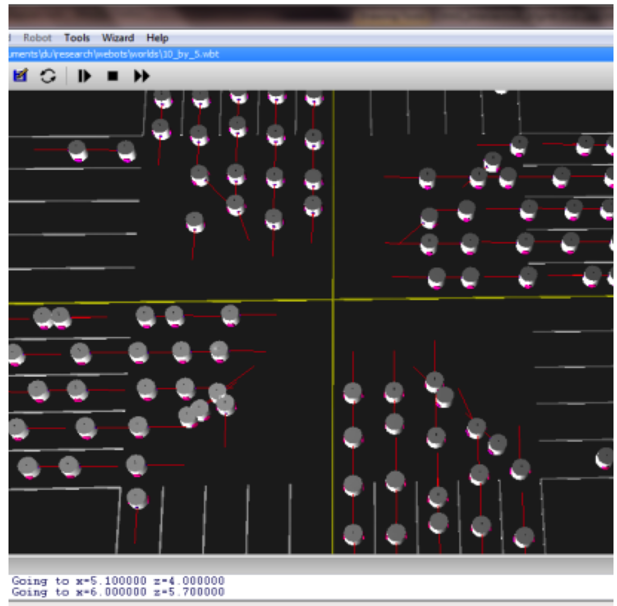

Figure 2: Screen shot of Webots simulations

on designing a robust and efficient distributed algorithm, we are focusing on the use of a simple discrete event simulator which allows us to represent the basic message exchange between agents in the system, and to scale the simulations efficiently to a large number of robots, and to evaluate a large number of parameter combinations. The result shown in Figure 1 come out of this simulation.

We have also utilized the Webots robot simulator which includes a more realistic physics engine so we can therefore evaluate the motion of the robots as well as the protocol. Figure 2 depicts a screen shot of one simulation we ran with robots lining up to enter the intersection. We have implemented the overpass, traffic-light and centralized reservation algorithms within Webots. Unfortunately, we are unable to scale these simulations sufficiently, but they allow for realistic integration with a robot platform including collision avoidance sensors and the like.

Finally, once we have progressed further with our distributed algorithm, we planned to implement the algorithm on a simple, indoor ground vehicle within a motion capture volume. In this scenario each vehicle will be running the algorithm locally, at the same time as it is running its basic control, communication, and sensing tasks. Evaluating this approach on real vehicles will give us a feel for the robustness of the solution, and the ability of vehicles to execute the protocol in real-time. A picture of the vehicles that we will use to do this is shown in Figure 3.

Future Research Directions

As mentioned previously, our work is ongoing. At present we are planning on pursuing the following.

- Improve the current peer-to-peer approach through the use of *vector clocks*. This will allow the robots to properly

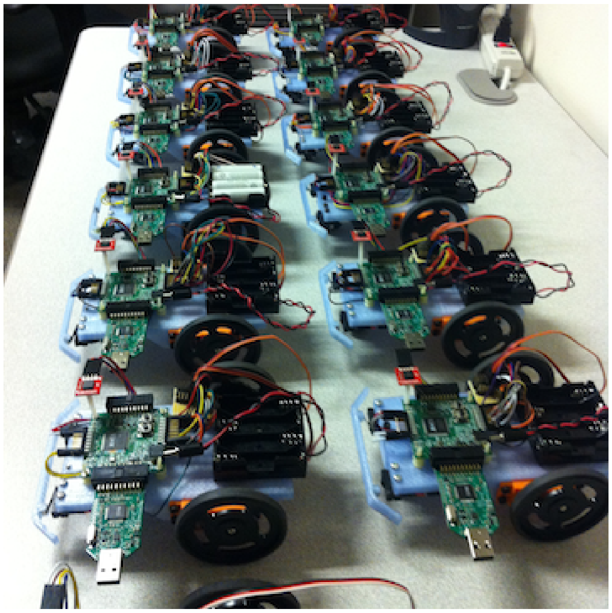

Figure 3: Team of mini ground vehicles to be used in physical evaluation.

order messages received and may reduce some of the inefficiency of the distributed approach.

- We are actively exploring the solution space provided by the existing simulations. This includes varying parameters having to do with the number of peers, the frequency of updates, and the conflict resolution function.
- We are considering evaluating different changes to the basic centralized algorithm, including allowing the reservation manager the ability to re-order reservations for better efficiency, canceling reservations that are causing problems, and such.
- We plan to relax constraints on the intersection and to include aerial vehicles through the addition of altitude. Specifically, we plan to allow vehicles to approach a shared waypoint from any heading, and to leave at any heading, and to support heterogeneous vehicle capabilities.
- We are working to increase the realism of simulation through dropped packets and varying latency. We are considering using the NS3 network simulator for a more faithful evaluation of network issues.
- We will be considering different decentralized approaches, including the use of a Distributed Hash Table to manage the reservations, hierarchical organization of robots (e.g., the robot at the end of the "line" is the group leader and can coordinate with the other group leaders).

References

Dresner, K., and Stone, P. 2004. Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '04, 530–537. Washington, DC, USA: IEEE Computer Society.

Dresner, K., and Stone, P. 2005. Multiagent traffic management: an improved intersection control mechanism. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, AAMAS '05, 471–477. New York, NY, USA: ACM.

Vogel, A.; Goerick, C.; and von Seelen, W. 2000. Evolutionary algorithms for optimizing traffic signal operation. In *Proceedings of the European Symposium on Intelligent Techniques*.